

**МІНІСТЕРСТВО КУЛЬТУРИ ТА
ІНФОРМАЦІЙНОЇ ПОЛІТИКИ УКРАЇНИ**

ХАРКІВСЬКА ДЕРЖАВНА АКАДЕМІЯ КУЛЬТУРИ

Факультет культурології та соціальних комунікацій

Кафедра цифрових комунікацій та інформаційних технологій

ПРОГРАМУВАННЯ ОФІСНИХ ДОДАТКІВ

Програма та навчально-методичні матеріали до курсу
для здобувачів першого (бакалаврського) рівня вищої освіти зі
спеціальності 029 «Інформаційна, бібліотечна та архівна справа»
ОП «Інформаційна та документаційна діяльність»

УДК 004.514(073+076.6)

П 78

Рекомендовано до видання науково-методичною радою ХДАК
(протокол № 13 від 29.04.2024 р.)

Рекомендовано кафедрою цифрових комунікацій та
інформаційних технологій
(протокол №15 від 26.03.2024 р.)

Рецензенти:

Брусенцев В. О., доцент, кандидат технічних наук,
доцент кафедри цифрових комунікацій
та інформаційних технологій ХДАК

Білова Т. Г., доцент, кандидат технічних наук,
доцент кафедри системотехніки

Харківського національного університету радіоелектроніки

Укладач:

Побіженко І. О., доцент, кандидат технічних наук,
доцент кафедри цифрових комунікацій
та інформаційних технологій ХДАК

П 78

Програмування офісних додатків : програма та навч.-метод.
матеріали до курсу для здобувачів першого (бакалавр.) рівня вищої
освіти зі спеціальності 029 «Інформаційна, бібліотечна та архівна
справа», ОП «Інформаційна та документаційна діяльність» / М-во
культури та інформ. політики України, Харків. держ. акад. культури,
Ф-т культурології та соц. комунікацій, Каф. цифрових комунікацій та
інформ. технологій ; [уклад. І О. Побіженко]. Харків : ХДАК, 2024. 51
с.

Навчальний курс «Програмування офісних додатків» є важливою складовою підготовки бакалаврів за спеціальністю 029 «Інформаційна, бібліотечна та архівна справа». Зміст курсу розкриває особливості однієї з суттєвих частин науки інформаційних технологій — програмування інструментальних засобів електронних документів. Основну увагу направлено на вивчення об'єктно-орієнтованих методів програмування на базі UNO-технології мовою ООО Basic.

УДК 004.514(073+076.6)

© Харківська державна академія культури, 2024 р.

© Побіженко І.О., 2024 р.

**ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
ПРОГРАМУВАННЯ ОФІСНИХ ДОДАТКІВ**

Опис навчальної дисципліни

Найменування показників	Галузь знань, напрям підготовки, освітньо-кваліфікаційний рівень	Характеристика навчальної дисципліни	
		денна форма навчання	заочна форма навчання
Кількість кредитів – 3	Галузь знань 0201. Культура	Вибіркова	
	Напрямок підготовки 029 «Інформаційна бібліотечна та архівна справа»		
	Загальна кількість годин - 90	Освітньо-професійна програма Інформаційно-документаційна діяльність	Рік підготовки:
4-й			
Тижневих годин для денної форми навчання: аудиторних 2 самостійної роботи студента – 4	Ступінь: бакалавр	Семестр	
		7-й	
		Лекції	
		10 год.	год.
		Семінарські	
		год.	год.
Практичні			
20 год.	год.		
Самостійна робота			
60 год.			

Примітка

Співвідношення кількості годин аудиторних занять до самостійної і індивідуальної роботи становить:

для денної форми навчання – 30/60

Мета та завдання навчальної дисципліни

Мета курсу – надання студентам теоретичних та практичних знань з сучасних технологій поширення згідно з потребами користувача функціональних можливостей офісних додатків операційних систем, так і їх використання у сучасній бібліотечно-інформаційній діяльності.

Загальні та фахові компетентності, які формує дисципліна.

Загальні компетентності	ЗК1) здатність до абстрактного мислення, аналізу та синтезу;
	ЗК2) здатність генерувати нові ідеї (креативність);
	ЗК3) вміння виявляти, ставити та вирішувати проблеми;
	ЗК4) здатність мотивувати людей рухатися до спільної мети;
	ЗК5) здатність виявляти ініціативу та підприємливість;
	ЗК6) здатність оцінювати та забезпечувати якість виконуваних робіт.
Професійні компетентності	ПК1) здатність до формування ефективної системи управління інформаційною, бібліотечною та архівною діяльністю;
	ПК3) здатність застосовувати технології для вирішення практичних, управлінських, науково-дослідних і прогностичних завдань у професійній діяльності
	ПК6) здатність застосовувати комп'ютерні технології з метою вдосконалення професійної діяльності, розуміти принципи проектування та функціонування автоматизованих БД, семантичного вебу, веб-сервісів та соціальних медіа;
	ПК9) здатність організовувати та реалізовувати маркетингову діяльність інформаційних установ;

	ПК12) здатність здійснювати інформаційний моніторинг
--	--

Програмні результати навчання (відповідно до освітньо-професійної програми).

Програмні результати навчання	ПРН1. Знати і розуміти наукові засади організації, модернізації та впровадження новітніх технологій в інформаційній, бібліотечній та архівній діяльності.
	ПРН2. Впроваджувати та використовувати комунікаційні технології у соціальних системах, мультимедійне забезпечення інформаційної діяльності, технології веб-дизайну та веб-маркетингу.
	ПРН8. Використовувати знання технічних характеристик, конструкційних особливостей, призначення і правил експлуатації комп'ютерної техніки та офісного обладнання для вирішення технічних завдань спеціальності.
	ПРН9. Оцінювати можливості застосування новітніх інформаційно-комп'ютерних та комунікаційних технологій для вдосконалення практик виробництва інформаційних продуктів і послуг.
	ПРН10. Кваліфіковано використовувати типове комп'ютерне та офісне обладнання.
	ПРН11. Здійснювати пошук інформації в різних джерелах для розв'язання професійних завдань.
	ПРН12. Застосовувати сучасні методики і технології автоматизованого опрацювання інформації, формування та використання електронних інформаційних ресурсів та сервісів.

Засоби оцінювання та методи демонстрування результатів навчання та підсумкова форма контролю

Засобами оцінювання та демонстрування результатів навчання та підсумкова форма контролю є:

- залік, іспит;
- тести;
- виконання практичних робіт;
- презентації результатів виконаних студентами практичних завдань.

Критерії оцінювання результатів навчання за 100 бальною рейтинговою шкалою:

1. Високий рівень знання матеріалу курсу (90 – 100 балів, А, відмінно):

вільне володіння понятійним апаратом,
вільна орієнтація в матеріалі курсу і здатність відповісти на різні питання;

високий рівень виконання програмних завдань.

2. Достатньо високий рівень знання матеріалу курсу (74 – 89 балів, С, В, відмінно):

добре володіння основними термінами і поняттями,
досить вільна орієнтація в матеріалі курсу і здатність відповісти на основні питання;

добрий рівень виконання програмних завдань.

3. Посередній рівень знання матеріалу курсу (60 -73 балів, D, E, задовільно):

посередній рівень володіння основними термінами і поняттями,

недостатньо вільна орієнтація в матеріалі курсу і здатність відповісти на окремі питання;

посередній рівень виконання програмних завдань.

4. Незадовільний рівень з можливістю повторного складання (35 – 59 балів, FX, незадовільно):

незнання окремих термінів та понять,

незнання основного матеріалу курсу і неспроможність відповісти на більшість питань;

незадовільний рівень виконання окремих програмних завдань.

5. Незадовільний рівень з обов'язковим повторним вивченням дисципліни (0 – 34 бали, F, незадовільно):

незнання основних термінів та понять,

незнання матеріалу курсу і неспроможність відповісти на питання;

нездатність до виконання більшості програмних завдань.

Основним критерієм успішності вивчення здобувачем/здобувачкою освіти навчальної дисципліни є отримання ним/нею не менш як 60 балів зі 100 за сукупністю усіх видів робіт: складання підсумкового іспиту, участь у лекційних заняттях, відповіді на семінарських заняттях, виконання завдань для самостійної роботи.

Кількість тижнів для теоретичної підготовки, кількість розділів та тем: 15 тижнів, 2 розділи та 7 тем.

ЗМІСТ І СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Структура навчальної дисципліни

Назви розділів	Кількість годин			
	Денна форма			
	Усього годин	у тому числі		
		Л	Лаб	С. р.
2	3	4	6	7
Розділ 1. Концепції програмування додатків OpenOffice.org.				
Суть засобів програмування офісних додатків операційних систем.	7	1		6
Основи програмування додатків мовою ООО BASIC.	9	1		8

Назви розділів	Кількість годин			
	Денна форма			
	Усього годин	у тому числі		
		Л	Лаб	С. р.
2	3	4	6	7
Концепція протокольного програмування макросів.	13	1	4	8
Концепція програмування макросів на рівні UNO-сервісів	13	1	4	8
Разом за розділом 1	42	4	8	30
Розділ 2. Засоби програмування електронних документів				
Робота з базами даних із середовища документів користувача.	16	2	4	10
Програмування текстових документів.	16	2	4	10
Програмування електронних таблиць.	16	2	4	10
Разом за розділом 2	48	6	12	30
Усього годин	90	10	20	60

ЗМІСТ КУРСУ

«Програмування офісних додатків» як навчальна дисципліна. Теоретичне та практичне значення курсу. Предмет, завдання, зміст і структура курсу, його місце у системі професійної підготовки фахівця з спеціальності 029 Інформаційна, бібліотечна та архівна справа. Взаємозв'язок курсу з іншими дисциплінами.

Джерелознавча база вивчення навчальної дисципліни: основна

і додаткова література.

РОЗДІЛ 1. КОНЦЕПЦІЇ ПРОГРАМУВАННЯ ДОДАТКІВ OPENOFFICE.ORG

Тема 1. Суть засобів програмування офісних додатків операційних систем.

Суть програмування додатків операційних систем та їх еволюція. Приклади типових завдань щодо поширення функціоналу додатків. Сучасні складові засобів програмування додатків **OpenOffice.org**: бібліотеки, модулі і макроси. Структура бібліотек. Структура модулів і макросів. Засоби користування бібліотеками, модулями і макросами.

Самостійна робота: Суть засобів програмування офісних додатків операційних систем

Тема 2. Основи програмування додатків мовою ООО BASIC.

Основні складові програми на **ООО BASIC**: суть процесу «оголосити, ввести, обробити та вивести» в умовах офісного програмування. Особливості застосування даних **ООО BASIC**: числових, строкових, дати і часу та об'єктних даних. Цикли та розгалуження. Поняття класу та об'єкту. Методи, властивості і події об'єктів. Синтаксис використання методів і властивостей об'єктів. Знайомство з меню подій і особливостями його застосування.

Самостійна робота: Основи програмування мовою **ООО BASIC**.

Тема 3. Концепція протокольного програмування макросів.

Поняття диспетчеру внутрішніх функцій і принцип його роботи. Загальні інструкції програмування диспетчера. Глобальні команди та їх використання у диспетчеру макросів. Технологія створення протокольного макросу засобами додатка **Open Office.org**. Загальна методика створення протокольного макросу програмуванням диспетчера.

Практичне заняття: Створення протокольних макросів у документах **OpenOffice.org**.

Самостійна робота: Організаційна структура засобів програмування електронних документів.

Тема 4. Концепція програмування макросів на рівні UNO-сервісів.

Основні поняття технології **UNO**: модулі, інтерфейси і сервіси та їх ієрархія. Поняття глобального сервісу **DeskTop** та його підпорядкувань: сервісів поточного компоненту, поточного фрейму та завантажника документів. Синтаксис файлового маршруту в концепції *OpenOffice.org*. Загальні сервісні засоби операцій з файлами. Поняття довідкової системи онлайн технології **UNO**, її основні функції і правила користування цією системою.

Практичне заняття: Знайомство з сервісами OpenOffice.org.

Самостійна робота: Основні поняття UNO-технології.

РОЗДІЛ 2.

ЗАСОБИ ПРОГРАМУВАННЯ ЕЛЕКТРОННИХ ДОКУМЕНТІВ

Тема 5. Робота з базами даних із середовища документів користувача.

Засоби вилучення інформації з бази даних у програму користувача. Програмні засоби реєстрації і підключення до бази даних. Засоби доступу до таблиць бази даних. Програмування запитів із застосуванням елементів мови **SQL**.

Практичне заняття: Робота з базами даних із середовища документів користувача засобами майстрів.

Практичне заняття: Робота з базами даних із середовища документів користувача засобами сервісів.

Самостійна робота: Засоби програмування взаємодій користувача з базами даних.

Тема 6. Програмування текстових документів.

Основний текстовий зміст, що представлено в інтерфейсах **XText**, **XTextRange** і **TextContent**. Засоби виділення, вставки і вилучення тексту. Форматування даних. Пошукові програмні засоби

текстових документів. Знайомство з основними властивостями та методами таблиць.

Практичне заняття: Організація обміну даними між текстовим документом і програмою користувача.

Практичне заняття: Робота з текстовими діапазонами.

Самостійна робота: Сервісні засоби UNO-технології для роботи з текстами.

Тема 7. Програмування електронних таблиць.

Засоби адресації до комірок та діапазонів. Програмування даних і формул. Програмування виділень, вилучень інформації із комірок і вставки даних у комірки. Форматування комірок.

Практичне заняття: Елементи управління в електронній таблиці.

Практичне заняття: Програмування електронної таблиці з елементами управління

Самостійна робота: Знайомство з елементами UNO-технології щодо програмування електронних таблиць.

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ДО ПРАКТИЧНИХ ЗАНЯТЬ І САМОСТІЙНОЇ РОБОТИ СТУДЕНТІВ

Теми практичних занять

№ з/п	Назва теми	Кількість годин
1	Створення протокольних макросів у документах OpenOffice.org.	4
2	Знайомство з сервісами OpenOffice.org	4
3	Робота з базою даних із середовища документа користувача засобами майстрів.	2
4	Робота з базою даних із середовища документа користувача засобами сервісів	2
5	Організація обміну даними між текстовим документом і програмою користувача.	2
6	Робота з текстовими діапазонами	2

7	Елементи управління в електронній таблиці..	2
8	Програмування електронної таблиці з елементами управління.	2
Разом		20

Теми самостійних робіт

№ з/п	Назва теми	Кількість годин
1	Суть засобів програмування офісних додатків операційних систем	6
2	Основи програмування мовою ООО BASIC.	8
3	Організаційна структура засобів програмування електронних документів	8
4	Основні поняття UNO-технології.	8
5	Засоби програмування взаємодій користувача з базами даних. .	10
6	Сервісні засоби UNO-технології для роботи з текстами..	10
7	Знайомство з елементами UNO-технології щодо програмування електронних таблиць.	10
Разом		60

Завдання до практичних занять та самостійної роботи здобувачів

Теми практичних занять

Практичне заняття № 1.

Тема: Створення протокольних макросів у документах OpenOffice.org

Мета: Навчитись створювати протокольні макроси у середовищі OpenOffice.org.

Завдання:

Опанувати методику створювати протокольних макросів, що підтримується інтерфейсом користувача.

Зміст та послідовність виконання заняття.

Створіть новий документ у середовищі текстового редактору Writer. Виконайте початок процедури створення протокового макросу засобами меню Сервіс/Макроси/Записати макрос.

Виконайте всі дії по створенню нової таблиці. Кількість рядків та стовпчиків завдайте самостійно. Закінчить процес створення таблиці командою Завершити запись. Створіть новий модуль у розділі Мої макроси вікна макроси Запишіть макрос, що створено, у розділ Мої макроси щигликом кнопки Записати і ознайомтесь з його текстом мовою ООО BASIC. Створіть засоби інтерфейсу користувача для запуску макросу, що було створено, засобами панелі інструментів, командою меню і комбінацією клавіш. Для цього скористайтесь кнопкою Назначити і відповідними засобами панелі Налаштування. Пред'явіть роботу викладачу та захистить.

Самостійно виконайте такі вправи:

Створіть протоковий макрос, що автоматично вставляє у текст графічний файл.

Створіть шаблон поздоровлення зі святом і відповідні засоби інтерфейсу користувача для вставки його у текст кнопкою панелі інструментів.

Створіть власний шаблон зі стилями і засоби інтерфейсу користувача для його призначення в електронному документі.

Коротко опишіть функціонал кожної інструкції диспетчера у макросах, що було створено.

Практичне заняття № 2.

Тема: Знайомство з сервісами OpenOffice.org

Мета: отримати концептуальні уявлення про взаємодію користувача з основними об'єктно-орієнтованими компонентами UNO-технології.

Завдання:

Опанувати методику роботи з головним сервісом UNO-технології Desktop.

Зміст та послідовність виконання заняття:

Ознайомтесь з головним сервісом UNO-технології Desktop.

Для цього відкрийте довідку он-лайн, як гіперпосилання. Створіть новий документ у середовищі OpenOffice.org, а у бібліотеці Standard макрос, текст якого запропоновано нижче.

Sub MyFrames

Dim oFrames As Object 'Для усіх фреймів

Dim oFrame As Variant *'Під окремий фрейм*

Dim i As Integer *'Індекс фрейму*

Dim s As String 'Для рядка друку

oDesktop = createUnoService("com.sun.star.frame.Desktop")

oFrames = oDesktop.getFrames()

'Створюємо цикл перебору усіх фреймів з індексами ' від нуля і далі до Count-1 та друкуємо їх заголовки.

For i = 0 To oFrames.getCount()-1 oFrame =

oFrames.getByIndex(i)

s = s & CStr(i) & " ." & oFrame.Title & CHR\$(10)

Next

MsgBox s, 0, "Заголовки фреймів"

End Sub

Запустіть цей макрос на виконання та перегляньте усі заголовки фреймів, що буде надруковано у MsgBox.

Відкрийте послідовно ще кілька документів за бажанням і

зверніть увагу на те, що при запуску цього макросу усі вони відображаються друком заголовків.

Доповніть цей макрос інструкціями, що надають інформацію про членів сервісу Desktop, встановив їх, наприклад, наприкінці процедури перед End Sub:

```

MsgBox
oDesktop.DBG_Properties
MsgBox
oDesktop.DBG_Methods
MsgBox
oFrames.DBG_Properties
MsgBox
oFrames.DBG_Methods

```

Запустіть знов макрос на виконання та зверніть увагу на довідку з методів та властивостей об'єктів. З окрема, властивість **DBG_Properties** надає повний список властивостей відповідного об'єкту, а властивість **DBG_Methods** — повний список методів. Упевніться у тому, що деякі з них застосовано у макросу, котрого було створено.

Навчіться тлумачити дані довідки. Для цього скористайтесь венгурською нотацією імен мов програмування. Наприклад, хай звісно, що ім'я **oDesktop** являє кореневий фреймовий сегмент додатка OpenOffice.org. Тоді метод **getFrames()** об'єкта **oDesktop** надає можливість залучити з середовища OpenOffice.org усі фрейми: документи, довідки додатку та макроси. У свою чергу метод **getCount()** об'єкта **oFrames** визначає скільки фреймів має завантаження додатка, а метод **getByIndex(i)** визначає той самий фрейм, що має значення індексу i. Властивість Title завжди надається під заголовки фреймів. Тип кожного з членів об'єкта у довідці доводиться. Наприклад, властивість Title приймає значення рядка символів, так що її можна зразу ж надрукувати у MsgBox. Більш детальна інформація про члени об'єктів UNO-технології надається з документації, що представлена на сайтах OpenOffice.org.

Розберіться з логікою програми макросу **MyFrames**, враховуючи подану вище інформацію.

Розгляньте альтернативні засоби адресації до документів: глобальні об'єктні змінні **StarDesktop** та **ThisComponent**. Першу з них призначено щоб представляти усе оточення документів OpenOffice.org, а другу — членів поточного документу. Ці члени можна переглянути засобами, що розглянуто вище. Для цього створіть та завантажте на виконання макрос **MyGlobeVar**:

Sub MyGlobeVar

MsgBox StarDesktop.DBG_Properties

MsgBox StarDesktop.DBG_Methods

MsgBox ThisComponent.DBG_Properties

MsgBox ThisComponent.DBG_Methods

End Sub

Випробуйте, як користуватись довідками на підпорядковані члени об'єктів. Так, наприклад, хай маємо для **StarDesktop** властивість **ActiveFrame**. Ясно, що ця назва має відношення до активного фрейму. Зліва від назви цієї властивості читаємо, що властивість **ActiveFrame** представляє собою об'єкт. Розгляньте у свою чергу її властивості. Їх може бути багато. Для цього, хай також у макросі **MyGlobeVar**, виконайте інструкцію

MsgBox StarDesktop.ActiveFrame.DBG_Properties

Далі читайте список властивостей **StarDesktop.ActiveFrame**. Серед них є, скажемо, така, що визначена, як **Title** і має тип рядок символів. Надрукуємо її значення у **MsgBox**:

MsgBox StarDesktop.ActiveFrame.Title

Завантажте цю інструкцію у макрос **MyGlobeVar** та запустіть макрос. Буде надруковано заголовок активного фрейму.

Випробуйте аналогічні вправи по користуванню іншими довідками UNO. У цьому ж напрямку можна поекспериментувати з об'єктом **ThisComponent**.

Створіть звіт з результатів виконання практичного заняття, де дайте відповіді на наступні питання:

1. Як адресувати до довідки он-лайн UNO-технології? Яку структуру має ця довідка?
2. Який із об'єктів OpenOffice.org являє собою кореневий фрейм? Як перелічити усі фрейми додатку OpenOffice.org та переглянути їх заголовки засобами програмування?
3. Як переглянути списки інтерфейсів, властивостей та методів об'єктів UNO?
4. Які області оперативної пам'яті визначаються об'єктами StarDesktop та ThisComponent?
5. Яку роль відіграють об'єкти StarDesktop та ThisComponent при програмуванні офісних документів?

Практичне заняття № 3.

Тема: Робота з базами даних із середовища документів користувача засобами майстрів

Мета: навчитися виконувати обмін даними між базами даних та документами користувача засобами майстрів.

Зміст та послідовність виконання заняття:

Створіть базу даних у середовищі **OpenOffice.org Base**, наприклад, Доходи від продажів, що має таке відношення: **Продажі(Код_продажу, Найменування_товару, Об'єм_продажів)**.

Подкачайте базу, що створено, даними так, щоб у таблиці Продажі було б хоча кілька рядків та закрийте базу.

Створіть новий текстовий документ.

У цьому документі створіть таблицю, що має формат, відповідний відношенню **Продажі**.

Додайте на екран панель інструментів Елементи управління з меню Від/Панелі інструментів/Елементи управління.

Перегляньте найменування кожного з елементів управління (інструментів). При вирішенні завдань з базами даних найбільший інтерес викликають елементи управління Список і Поле із списком. Ці інструменти підтримуються майстрами дизайну форми.

Зверніть увагу на інструмент Режим розробки. Цей інструмент дозволяє перемикатися з режиму розробки форми на режим виконання.

У режимі розробки перетягніть інструмент Список у поле документа так, щоб при розгляді форми зображувався стовпець **Найменування_товару**.

В процесі перетягування Списку відповідайте по сенсу на питання майстра відносно того, звідки брати дані і куди їх пересилати. Цей процес для користувача достатньо прозорий.

Встановіть властивості елементів управління Список у стан Лише для читання. Це для того, щоб випадково не пошкодити базу даних.

Повторіть аналогічні дії з інструментом Список так, щоб зобразити стовпець **Об'єм_продажів**.

Перейдіть в режим виконання форми, клацнувши інструмент Режим розробки.

Повинна автоматично зобразиться інформація з таблиці Продажі бази даних. На жаль, копіювання даних користувачем з отриманих списків майстром не передбачено. Синхронізація виділень також полягає на користувача. Але, вся необхідна для роботи інформація є. Після того, як ця інформація буде оброблена користувачем, елементи управління можна видалити в режимі розробки форми.

Виконайте усі завдання, що маєте вище, засобами майстра елемента управління Поле із списком і впевніться в їх дії.

Навчитись користуватись засобами команди Вид/Джерела даних. Для цього спочатку надайте цю команду. Потім засобами провідника, що відкриється у відповідь, виберіть ту базу даних, що потрібно, скажемо, Доходи від продажів, а у контекстному меню виберіть команду **Копіювати**. У відповідь відкриється вікно **Вставка** стовпчиків бази даних. Це вікно має прозорий інтерфейс користувача. Воно дозволяє скопіювати у поточний документ усю таблицю або її

окремі поля або текст з будь-якого поля таблиці бази даних.

Засобами вікна Вставка стовпчиків виконайте копіювання даних з бази даних Доходи від продажів у таблицю, що створено у текстовому документі.

Створіть нову електронну таблицю.

Виконайте усі завдання, що зазначено вище, також для документа Calc та упевнитися у тому, що процес обміну даними виконується.

Створіть звіт з результатів виконання практичного заняття, де дайте відповіді на наступні питання:

1. Які елементи управління застосовуються для взаємодії з базою даних з іншого документа?
2. Як користуватися майстром для вилучення інформації з бази даних засобами елементів управління?
3. Яку команду застосовують для можливостей обміну даними між документом користувача і базою даних?
4. Як користуватися командою Вид/Джерела даних щоб мати можливість скопіювати інформацію з бази даних у документ користувача?

Практичне заняття № 4.

Тема: Робота з базами даних із середовища документів користувача засобами сервісів

Мета: навчитися виконувати обмін даними між базами даних та документами користувача засобами сервісів.

Завдання. Створити та випробувати програмні засоби користувача щодо перегляду структури бази даних та наповнення її таблиць даними.

Зміст та послідовність виконання заняття:

Створіть нову базу даних, з двома відношеннями:

Продажі(Код_продажу, Найменування_товару, Об'єм_продажів); Клієнти(Код_клієнта, Компанія, Місто).

Обов'язково зареєструвати базу у СУБД. Можна скористатися і таким чином: додавати у базу, що створена на попередньому занятті, таблицю **Клієнти**. Для спрощення процесу створення такої таблиці рекомендується задіяти **Майстра**.

Подкачайте кожен з таблиць кількома рядками даних і закрийте базу.

Навчитися переглядати структуру бази даних з програми користувача. Для цього у середовищі будь-якого документу створіть макрос, який, скажемо, буде мати ім'я **MyBase**. Ми будемо розглядати його як головну процедуру у обраному модулі.

Поперед початку макросу об'явить змінну, яку можна буде побачити з усіх процедур макросу **MyBase**.

Dim oDb As Object ' Це буде об'єкт бази даних

Для підключення до бази даних створіть у тілі модуля, де визначено **MyBase** окрему процедуру, котра і буде виконувати дії підключення до бази даних. Призначимо ім'я такої процедури як **BaseConnection**. Складемо текст програми згідно з концепцією **UNO**-технології:

Sub BaseConnection(dbName as String)

'Виконаємо підключення до бази:

dbContext =

createUnoService("com.sun.star.sdb.DatabaseContext")

oDataSource = dbContext.getByname(dbName)

oDb=oDataSource.GetConnection("", "")

End Sub

У даному тексті передбачено з метою підключення до бази даних пересувати ім'я цієї бази через список параметрів. Само підключення відбувається застосуванням **UNO**-сервісу **DatabaseContext**. Цей сервіс автоматично обере відповідний драйвер бази завдяки методу **getByName(dbName)**. Остаточне підключення відбувається відпрацюванням метода **GetConnection("", "")**. При викликах цього

метода можуть бути задіяні логін та пароль, що наводяться у лапках. У спрощеному варіанті макросу припущено, що вони розглядаються, як пусті.

Виконайте налагодження цієї процедури підключення. Поперед усього перевірте, чи відбувається підключення до створеної бази даних. Для цього запустіть на виконання макрос **MyBase**, в якому, поки ще, передбачимо єдину інструкцію:

BaseConnection("Доходи від продажу")

Якщо маєте помилки, що визначено відповідними повідомленнями, то виправте усі їх.

Якщо помилок не зроблено, то процес виклику скінчиться без повідомлень.

Перегляньте назви таблиць бази даних. Для утворення такого перегляду треба спочатку знайти метод об'єкта **oDb**, що надає можливість доступу до всіх таблиць. Якщо роздрукувати, наприклад, у вікні **MsgBox** методи об'єкта **oDb**, то можна побачити там метод **getTables**, що позначено як об'єкт. Цей об'єкт визначає доступ до усіх таблиць бази даних. Якщо переглянути методи цього об'єкта, можна побачити, що він, у свою чергу використовує метод **getElementNames**. Цей метод позначено як масив рядків символів, де кожен елемент має ім'я таблиці бази даних. Загальна кількість таких рядків визначається властивістю **Count** об'єкта **dbTables**.

З урахуванням усього цього, процедура перегляду таблиць може мати, наприклад, такий вигляд:

Sub AllTables()

Dim dbTables as Object

Dim sTableNames() as

String Dim i as Integer

dbTables = odb.getTables()

sTableNames = dbTables.getElementNames

'Створюємо цикл по іменам

таблиць **For i=0 to**

dbTables.Count -1

```
MsgBox(sTableNames(i))
```

```
Next i
```

```
End Sub
```

Помістіть цей текст у модуль, де розташований макрос **MyBase**.

Запишіть після інструкції підключення до бази даних інструкцію виклику друку імен таблиць: **AllTables()** і запустить макрос **MyBase** на виконання. Якщо помилок не зроблено, то вікно **MsgBox** з'явиться стільки раз, скільки ми маємо таблиць і кожен раз буде зображено ім'я відповідної таблиці.

Створіть програмні засоби перегляду даних з таблиць. Для цього застосуйте сервіс **RowSet**, який відпрацьовує за алгоритмом: поєднання з підключенням бази даних, завдання команди у форматі **SQL** на вибірку даних, запуск **RowSet**. Внаслідок таких дій усі дані, що замовлено у **SQL**, буде втілено у об'єкт **RowSet**. Для перегляду рядків можна застосувати метод **Next** цього ж об'єкту.

Приклад процедури, що виконує перегляд даних, де команда **SQL** передається через параметри процедури дано нижче:

```
Sub dbRows(sSQL as String)
```

```
Dim i as Integer
```

```
Dim oRowSet as Object
```

```
Dim sMyRow as String
```

```
oRowSet= createUnoService("com.sun.star.sdb.RowSet")
```

```
oRowSet.activeConnection= oDb 'Активуємо з'єднання
```

```
oRowSet.Command = sSql 'Надаємо команду SQL
```

```
oRowSet.execute 'Виконуємо команду SQL
```

```
'Друкуємо результати
```

```
Do while oRowSet.Next
```

```
For i=1 to oRowSet.Columns.Count
```

```
sMyRow= oRowSet.getString(i)
```

```
msgbox(sMyRow)
```

```
next i
```

```
Loop
```

```
End Sub
```

Цей текст також розмістіть у модулі, де розташований макрос **MyBase**, а у самому макросі створіть команду **SQL** під вибірку даних з якоїсь із таблиць. Потім виконайте виклик процедури **dbRows**. Приклад усього тексту процедури макросу **MyBase** наведено нижче:

```
Dim oDb As Object
```

```
Sub MyBase
```

```
Dim sSQL as String
```

```
BaseConnection("Доходи від продажів")
```

```
AllTables()
```

```
sSql="SELECT""Код_продажі"","""Найменування  
товару"",""
```

```
+_
```

```
""""Об'єм продаж (грн.)"" FROM ""Продажі """"
```

```
dbRows(sSql)
```

```
End Sub
```

Для спрощення процесу створення команди **SQL** можна поступити так. Спочатку відкрийте базу даних, потім створіть в ній запит на вибірку та упевнитись в тому, що він добре працює. В решті решт розгляньте свій запит у вікні **SQL**, скопіюйте його і закрийте базу. Помістіть цей фрагмент у макрос, але майте на увазі, що поперше текст команди подається у лапках, а по-друге

— якщо у цьому тексті є послідовності символів, котрі повинні бути теж у лапках, то для них застосовуються подвійні лапки. У наведеному прикладі процедури **MyBase** це зображено.

Знову запустіть **MyBase** на виконання. Якщо все зроблено вірно, то спочатку будуть роздруковані імена усіх таблиць бази даних, а потім — усі рядки з таблиці, що замовлено у запиті.

Створіть звіт з результатів виконання практичного заняття, де дайте відповіді на наступні питання:

1. Як можна виконати підключення до бази даних, що зареєстрована у СУБД?
2. Який метод об'єкта бази даних дає можливість отримати інформацію з переліку таблиць бази даних ?

3. Який сервіс забезпечує доступ до усіх кортежів бази даних?
Який формат має команда **SQL** на вибірку даних з таблиці?
4. За яким алгоритмом відпрацьовують сервіс RowSet для перегляду даних з таблиці?

Практичне заняття № 5.

Тема: Організація обміну даними між текстовим документом і програмою користувача .

Мета: навчитися типовим прийомам організації обміну даними між текстовими документами та програмами користувача.

Завдання. Опанувати програмні засоби користувача щодо вставки та змінення тексту текстових документів.

Зміст та послідовність виконання заняття:

Створіть новий текстовий документ.

Засобами створення макросів створіть новий макрос у розділі нового документа:

Sub CreateInsertText

Dim oText as Object

Dim oCursor as Object

Dim oDocument as Object

oDocument=thisComponent

oText = oDocument.Text

oCursor = oText.createTextCursor()

oText.insertString(oCursor,"З Новим XXXX роком! ", False) End

Sub

Запустить макрос на виконання. Згідно з інструкціями цього макросу у текстовому документі повинно бути надруковано рядок символів:

З Новим XXXX роком!

Виконайте налагодження макросу так, щоб цей друк відбувся.

Удоскональте текст макроса так, щоб було замість символів «XXXX» надруковано поздоровлення з конкретним роком. Для цього макрос продовжимо такою послідовністю інструкцій:

NewYear = InputBox("Введіть який буде наступний рік")

oCursor.gotoEnd(False)

Пересунемо курсор у кінець тексту

oCursor.gotoPreviousWord(false) 'Повернемось на слово назад
oCursor.gotoPreviousWord(true) 'Повернемось на слово назад
oText.insertString(oCursor, NewYear + " ", true) 'Друкуємо
 зміни

Видалити весь попередній текст документа і запустити макрос, що продовжено, на виконання.

Впевніться у тому, що коли будете вводити, наприклад, 2014, то спочатку роботи макросу треба надрукувати рядок символів:

З Новим XXXX роком!

А зразу ж після вводу у **InputBox** нових даних цей рядок буде виправлено:

З Новим 2024 роком!

Добре розберіться, коли відбуваються заміни тексту, а коли зміни тексту з продовженням. Для цього виконайте створений макрос чотири рази підряд так, щоб у останніх двох інструкціях: інструкції пересування курсору і інструкції вставки тексту відбулися зміни специфікації **bExpand** з таким чергуванням:

False, False

False, True

True, False

True, True.

Кожного разу перед запуску макросу удаляйте увесь текст з документу.

При тлумаченні результатів заміни тексту намітьте, що добуток двох логічних змінних є істиною тільки тоді, коли обоє змінні є істинні.

Створіть звіт з результатів виконання практичного заняття, де дайте відповіді на наступні питання:

1. Як програмно адресуватися до тексту поточного документа засобами **UNO**-технології?
2. Як адресуватися до сервісу, що призначено для пересувань по тексту? Чи зображається модель невидимого курсора на екрані монітора? Яка позиція тексту визначається при

- створенні моделі текстового курсору?
3. Який метод застосовується для вставки тексту у позицію, що визначено курсором, як об'єктом?
 4. Як визначається сенс пари специфікацій bExpand з пересувань та вставки?

Практичне заняття № 6.

Тема: Робота з текстовими діапазонами.

Мета: навчитися типовим прийомам роботи з текстовими діапазонами.

Завдання. Опанувати програмні засоби користувача щодо вставки та змінення тексту документів у діапазоні за визначенням.

Зміст та послідовність виконання заняття:

Створіть новий текстовий документ. Запишіть у цей документ кілька абзаців тексту. Засобами створення макросів створіть новий макрос у розділі нового документа:

Dim oDocument as Object

Dim oText as Object

Dim oVCursor as Object

Dim oCursor as Object

Dim str as String

oDocument = thisComponent

oText = oDocument.Text

oVCursor =

oDocument.getCurrentController().getViewCursor()

oCursor = oText.createTextCursorByRange(oVCursor)

oCursor.gotoStartOfParagraph(false)

oCursor.gotoEndOfParagraph(true)

str = oCursor.getString

MsgBox(str)

End Sub

Зверніть увагу на інструкцію

oVCursor

=

oDocument.getCurrentController().getViewCursor()

Дана інструкція створює об'єкт, що визначає видимий курсор. Ми будемо застосовувати її для управління моделлю невидимого курсора **oCursor** при роботі з діапазонами тексту. Об'єкт невидимого курсора, котрий пов'язується з видимим курсором, створюється за інструкцією

oCursor = oText.createTextCursorByRange(oVCursor)

Методи-функції **gotoStartOfParagraph** та **gotoEndOfParagraph** надають можливість визначити початок та кінець того абзацу, де відбувся невидимий курсор. Але ж коли його його пов'язано з видимим курсором, це надає більшої комфортності програмуванням. Запустить макрос на виконання та впевнитися в тому, що у вікні **MsgBox** відображається весь абзац, котрий вміщує в себе курсор. Змінити текст абзаца, що було виділено будь-яким текстом. Для цього рекомендується застосувати інструкцію, знайомство з якою відбулось на попередньому занятті:

oText.insertString(oCursor, String, true)

Цю інструкцію розмістіть у тексті макросу замість **MsgBox(str)**. Параметр **String** слід завдати як строковий вираз, або як строкову змінну.

Завдайте таку строкову змінну за бажанням і запустить макрос. Абзац повинно бути замінено на текст строкової змінної.

Навчіться видаляти абзаци, що визначено видимими курсорами, програмними засобами. Видалення абзаців також виконується за наведеним фрагментом програми: достатньо параметр **String** завдати пустим значенням "".

Опануйте перегляд абзаців. Для таких дій організуйте цикл, котрий буде виконуватись до тих пір, поки наступний абзац існує. В цьому випадку маємо **oCursor.gotoNextParagraph(false) = true**.

Тому цикл перегляду абзаців можна організувати, наприклад, так:

Do while oCursor.gotoNextParagraph(false)
oCursor.gotoEndOfParagraph(True)

str = oCursor.getString
MsgBox(str)

Loop

Включить цей фрагмент у кінець макросу, установить курсор у будь-який абзац документу за бажанням і запустить макрос на виконання. Якщо усе виконано без помилок, то кожного разу до кінця документа у вікні **MsgBox(str)** буде зображатись текст чергового абзацу. Програму налагодьте, якщо були помилки. Потурбуйтеся при цьому, щоб текст першого абзацу видимого курсору не змінювався, не видалявся, а читався.

Створить звіт з результатів виконання практичного заняття, де дайте відповіді на наступні питання:

1. Які програмні засоби **UNO**-технології застосовуються для переміщенням по абзацам текстового документа згідно з довідкою
www.openoffice.org/api/docs/common/ref/com/sun/star/text/XParagraphCursor.html ?
2. Як слід завдавати вхідні параметри та отримувати вихідні значення методів інтерфейсу **XParagraphCursor**?
3. Як пов'язати видимий та невидимий курсори?
4. Як програмно змінити текст абзацу на інший?
5. Як організувати програмне видалення тексту з абзацу?
6. Як створити цикл для перегляду всіх абзаців поточного документа?

Практичне заняття № 7.

Тема: Елементи управління в електронній таблиці.

Мета: навчитися взаємодіяти з діапазонами комірок та елементами управління при програмуванні завдань, котрі представлені на кількох аркушах.

Завдання. Засвоїти застосування елементів управління в електронних таблицях на прикладі створення проекту електронної таблиці **Закази** урахуванням таких вихідних даних:

Прейскурант цін на товари розміщується на першому аркуші і має наступний вигляд:

	A	B	C	D
1	Прейскурант цін			
2	№ п/п	Найменування товару	Одиниця виміру	Ціна за одиницю
3				
4				
5				

Інтерфейс користувача проекту (касира) розміщується на другому аркуші і має зображати наступне:

Цей інтерфейс повинен забезпечити такі можливості проекту.

Зі списку **Вибір з найменувань товару** касир обирає найменування того поточного товару, якого було замовлено.

У комірці **D5 (Ціна)** повинна автоматично зображуватись ціна товару, що обрано на замовлення.

Комірку **E5 (Кількість)** касир заповнює по замовленню. Сума замовленого товару повинна зобразитись автоматично у комірці **F5 (Сума)**.

Щиглик кнопки **У корзину** повинен зобразити загальну суму товару, що замовлено. Ця сума повинна зобразитись у комірці **F7**.

Кнопка **На рахунок** повинна встановити інтерфейс користувача у початкове становище, підготував його до прийому нового замовлення.

Рахункова частина проекту повинна бути розміщеною на

третьому аркуш і мати вигляд, як показано на примірнику нижче:

A	B	C	D	E	F
Замовл. №	5				
№ п/п	Найменування товару	Одиниця виміру	Ціна за одиницю	Кількість товару	Сума (Грн.)
1	Сметана	кг	15,00 грн.	2	30,00 грн.
2	Сир	кг	80,00 грн.	2	160,00 грн.
3	Молоко	л	7,00 грн.	3	21,00 грн.
				Усього:	211,00 грн.
Замовл. №	6				
№ п/п	Найменування товару	Одиниця виміру	Ціна за одиницю	Кількість товару	Сума (Грн.)
1	Кефір	л	8,00 грн.	3	24,00 грн.

Функціонал кнопок **У корзину** та **На рахунок** у рахунковій частині проекту повинен забезпечити автоматичне формування рахунків.

Порядок виконання завдання.

Уважно ознайомтесь з завданням та добре з'ясуйте, що треба зробити у проекті.

Створіть нову книгу у середовищі **Calc** й підготуйте на першому аркуші прейскуронт цін за бажанням, але так щоб там було порядку десятка найменувань товарів.

Відкрийте наступний аркуш і створіть інтерфейс користувача так, щоб комірки отримали наступні формати: **D5, F5, F7** — грошові, **E5** — числові, а решта — текстові.

Доповніть інтерфейс користувача елементами управління: списком, та двома кнопками, згідно завдання.

Завдайте списку такі властивості.

Властивості: властивості. Вкладка -Дані:

Пов'язана комірка: Н3,

Зміст пов'язаної комірки: Позиція виділеної записи,

Вихідний діапазон комірок: Лист1:В3:В8.

Стовпчик **Н** використовуйте як метадані. Комірку **Н3** виберіть як

посередника між преїскурантом цін і елементом управління **Список** таким чином, щоб її значення завжди вказувало на позицію запису у преїскуранті після того, як буде обрано товар у списку. Тому, якщо треба, виправте вказаний на примірнику діапазон **Лист1.В3:В8** згідно з числом поданих у преїскурант найменувань.

Відповідно до цього запрограмуйте комірку **D5** так, щоб автоматично бачити у ній зображення ціни товару з преїскуранту цін:
=VLOOKUP(H3;Лист1.A3:D8;4)

Специфікацію функції **VLOOKUP** обов'язково розгляньте у довідці до **Calc**.

Введіть у комірку **F5** формулу: **=D5*E5** для обчислення суми замовлення на поточний товар.

Виконайте налагодження інтерфейсу проекту так, щоб при виборі продукту зі списку найменувань у стовпчику **Сума** було зображено добуток ціни по преїскуранту на кількість замовленого товару. При цьому, якщо кількість товару не завдана, то й було б зображено нульову суму.

Збережіть увесь документ в особистій теці. Його програмування буде продовжено на наступному занятті.

Створіть звіт з результатів виконання практичного заняття, де дайте відповіді на наступні питання:

1. Як встановити панель елементів управління на екрані монітору?
2. Як звернутися до властивостей елемента управління та встановити потрібні?
3. Як налаштувати елемент управління **Список** на дані з аркуша електронної таблиці?
4. У чому є призначення функції **VLOOKUP** в умовах створення проектів на кількох аркушах?

Практичне заняття № 8.

Тема: Програмування електронної таблиці з елементами управління

Мета: отримати уявлення про програмування електронних таблиць з елементами управління при створенні проектів, котрі потребують декількох аркушів.

Завдання. Опанувати методику програмування виконавчої частини проекту Закази, інтерфейс якого було здійснено на попередньому занятті. При цьому реалізувати функціонал виконавчої частини проекту Закази такими складовими:

- Ініціалізація проекту;
- Заповнення кошика товарів на замовлення покупця;
- Формування рахунку на кошик товарів.

Зміст та послідовність виконання заняття.

Відкрийте проект Закази, що створено на попередньому практичному занятті. В цьому проекті створіть новий модуль, де будете розміщувати відповідні макроси.

Ініціалізацію проекту виконайте у такій послідовності. Створіть макрос, котрий при кожному відкритті проекту Закази буде очищати усю частину рахунків, що розташована на третьому аркуші. Такий макрос дуже просто виконати засобами диспетчера. Для цього слід запротоколювати виділення та очищення усього третього аркуша клавішами **Ctrl/A, Delete**, та скопіювати текст макросу, що надано диспетчером у свій модуль.

Назвіть процедуру макросу **ClearSheet3**. Перевірте, як спрацює ця процедура на третьому аркуші, коли у полі цього аркушу дані завдані у комітках, що обрано випадково. Якщо усе добре, переходьте до наступного кроку. Якщо – ні, то поладуйте процес.

Процедуру макросу **ClearSheet3** на першому місті залишати не слід, бо вона є тільки підсобною і буде заважати розглядати весь проект з початку. Тому у будь-який зручний момент можна перетягнути її у самий кінець програмування проекту.

Позначте ім'я процедури ініціалізації проекту: **InitForm**.

Програмування цієї процедури виконайте у такій послідовності, як дано нижче:

Sub InitForm()

' Для кнопки на панелі інструментів

' Оголосимо дані

Dim oSheet1 as Object ' Аркуш1

Dim oSheet2 as Object ' Аркуш2

Dim oSheet3 as Object ' Аркуш3

Dim oCell1 as Object ' Комірки аркушу 1

Dim oCell2 as Object ' Комірки аркушу 2

Dim oCell3 as Object ' Комірки аркушу 3

Dim j as Integer ' Робоча змінна

Dim volS as String

' Робоча змінна

' Очищуємо третій аркуш

Call ClearSheet3

' Получимо аркуш 3, індексація аркушів починається з нуля.

oSheet3=ThisComponent.Sheets(2) ' Получимо аркуш 2

oSheet2=ThisComponent.Sheets(1)

' Заповнюємо комірки стовпчику H (Пояснення. нижче)

oCell2=oSheet2.getCellByPosition(7, 1) oCell2.value=1

oCell2=oSheet2.getCellByPosition(7, 2) oCell2.value=0

oCell2=oSheet2.getCellByPosition(7, 3) oCell2.value=1

oCell2=oSheet2.getCellByPosition(7, 4) oCell2.value=0

' Друкуємо на третьому аркуші "Замовлення № " і значення 1

oSheet3=ThisComponent.Sheets(2)

oCell2=oSheet2.getCellByPosition(7, 1)

oCell3=oSheet3.getCellByPosition(0, 0)

oCell3.string="Замовлення № "

oCell3=oSheet3.getCellByPosition(1, 0) oCell3.value=1

' З першого аркуша переносимо на третій аркуш усі заголовки шапки

oSheet1=ThisComponent.Sheets(0) For j=0 to 3

```

oCell1=oSheet1.getCellByPosition(j, 1)
oCell3=oSheet3.getCellByPosition(j, 1) volS=oCell1.string
oCell3.string=volS next j
' Додаємо на шапку заголовки: "Кількість" і "Сума (Грн.)"
oCell3=oSheet3.getCellByPosition(4, 1)
oCell3.string="Кількість"
oCell3=oSheet3.getCellByPosition(5, 1) oCell3.string="Сума
(Грн.)"
' Пишемо значення «0» у комірку E5 другого аркушу
oCell2=oSheet2.getCellByPosition(4, 4) oCell2.Value=0
' Ширину стовпчиків аркушу3 оптимізуємо
oSheet3.setColumns.OptimalWidth=true
End Sub

```

Нижче надано сенс заповнення метаданими стовпчика Н другого аркуша:

Н	І
1	Номер поточного замовлення
0	Номер рядка, котрий пов'язано зі списком
1	Індекс останнього рядка на третьому аркуші
0	Номер товару у корзині

Випробуйте макрос простим завантаженням на виконання. Якщо усе зроблено вірно, то на другому аркуші з'являться дані у стовпчику Н, а на третьому аркуші буде роздрукована шапка замовлення №1. У противному випадку налагодьте процедуру InitForm.

Макросу InitForm визначте відповідну кнопку на панелі інструментів та випробуйте її у дії.

Запрограмуйте кнопку У корзину. Для цього спочатку створіть макрос toBasket у послідовності, що надано нижче:

```

Sub toBasket
' Для кнопки У корзину
Dim oSheet1 as Object
Dim oSheet2 as Object

```

Dim oSheet3 as Object

Dim oCell1 as Object

Dim oCell2 as Object

Dim oCell3 as Object

Dim j as Integer

Dim NumString as Integer

Dim NumProduce as Integer

Dim NumList as Integer

Dim sum as Currency

Dim volS as String

' Початок коригування стовпчику Н та відповідних змінних.

' Читаємо з комірки Лист2.Н4 значення номеру рядка (NumString)

' що було заповнено в останню чергу:

oSheet2=ThisComponent.Sheets(1)

' Комірки індексуються з нуля, тому маємо індекси (7, 3)

oCell2=oSheet2.getCellByPosition(7, 3)

NumString=oCell2.value

' збільшуємо номер останнього рядка на одиницю

' щоб мати поточний індекс рядка NumString у третьому аркуші.

NumString=NumString+1 oCell2.value=NumString

' Корегуємо цей номер у комірки Н4.

oCell2=oSheet2.getCellByPosition(7, 3)

oCell2.Value=NumString

' Аналогічні дії виконуємо з номером товару у корзині Лист2.Н5

oCell2=oSheet2.getCellByPosition(7, 4)

NumProduce=oCell2.value

NumProduce=NumProduce+1 oCell2.value=NumProduce

' Читаємо номер рядка у преїскуранті цін з комірки Лист2.Н3

' збільшуємо його ще на одиницю, щоб мати готовий індекс рядка

' при адресації до рядків першого аркушу:

oCell2=oSheet2.getCellByPosition(7, 2)

Numlist=oCell2.value+1

' Кінець коригування стовпчику Н та відповідних змінних.

```

' Пишемо всю інформацію о товарі у корзину на третій аркуш.
' Адресуємо до третього аркушу по індексу рядка NumString
' У цей рядок пишемо номер товару (NumProduce)
oSheet3=ThisComponent.Sheets(2)
oCell3=oSheet3.getCellByPosition(0, Numstring)
oCell3.value=NumProduce
' Три наступні стовпчики заповнюємо з першого аркушу
аналогічно:
oSheet1=ThisComponent.Sheets(0)
For j=1 to 3
  oCell1=oSheet1.getCellByPosition(j, NumList)
  oCell3=oSheet3.getCellByPosition(j, Numstring)
  volS=oCell1.string
  oCell3.string=volS
next j
' Заповнюємо комірку Кількість
oCell2=oSheet2.getCellByPosition(4, 4)
oCell3=oSheet3.getCellByPosition(4, Numstring)
oCell3.Value=oCell2.Value ' та комірку Сума з другого аркушу
oCell2=oSheet2.getCellByPosition(5, 4)
oCell3=oSheet3.getCellByPosition(5, Numstring)
sum=oCell2.Value
oCell3.Value=oCell2.Value
' Збільшуємо загальну суму заказу на другому аркуші
' на суму поточного товару, що у корзині:
oCell2=oSheet2.getCellByPosition(6, 10)
oCell3=oSheet3.getCellByPosition(6, Numstring)
oCell2.Value=oCell2.Value + Sum
End Sub

```

Призначте цей макрос кнопці У корзину. Для цього у режимі розробки проекту розкрийте вікно даної кнопки і, потім розкривши список макросів кнопкою з кількома крапками справа від події Призначити дію, виберіть: toBasket.

Упевніться, що макрос працює вірно. Для цього послідовно виберіть кілька найменувань товарів зі списку, та почергово відправте замовлення у корзину. На третьому аркуші повинні відобразитись усі замовлення. Але загальні підсумки поки що відсутні. Якщо не так, то виконайте налагодження макросу.

Запрограмуйте кнопку На рахунок. Для цього спочатку створіть макрос Oder у послідовності, що надано нижче:

Sub Order

' Для кнопки На рахунок

Dim oSheet1 as Object

Dim oSheet2 as Object

Dim oSheet3 as Object

Dim oCell1 as Object

Dim oCell2 as Object

Dim oCell3 as Object

Dim oRange2 as Object

Dim oRange3 as Object

Dim j as Integer

Dim NumString as Integer

Dim volD as Double

Dim volS as String

' Закінчити оформлення поточного замовлення

oSheet2=ThisComponent.Sheets(1)

oSheet3=ThisComponent.Sheets(2)

oCell2=oSheet2.getCellByPosition(7, 3)

NumString=oCell2.value NumString=NumString+1

oCell2=oSheet2.getCellByPosition(7, 3)

oRange2=oSheet2.getCellRangeByName("C11:G11")

oRange3=oSheet3.getCellRangeByPosition(1,NumString,5,NumString)

oRange3.setDataArray(oRange2.GetDataArray())

' Обнулити загальну суму заказу на наступне замовлення

oCell2=oSheet2.getCellByPosition(6, 10)

oCell2.value=0

'Обнулити номер товару у корзині

oCell2=oSheet2.getCellByPosition(7, 4)

oCell2.value=0

'Получити з метаданих другого аркуша індекс Numstring останнього ' рядка на третьому аркуші і збільшити його на 4, ' щоб було видно розподіл проміж замовленнями.

oCell2=oSheet2.getCellByPosition(7, 3)

NumString=oCell2.value

NumString=NumString+4

oCell2.value=NumString

' На третьому аркуші записати рядок символів"Заказ №"

oCell3=oSheet.getCellByPosition(0, Numstring)

oCell3.string="Заказ № "

'Прочитати із метаданих другого аркушу номер поточного ' замовлення і збільшити його на одиницю.

oCell2=oSheet2.getCellByPosition(7, 1)

oCell2.value=oCell2.value+1

' Записати цей номер на третьому аркуші:

oCell3=oSheet3.getCellByPosition(1, Numstring)

oCell3.Value=oCell2.value

'Відобразити номер нового замовлення у метаданих:

oSheet1=This- Component.Sheets(0)

Numstring=Numstring+1

oCell2=oSheet2.getCellByPosition(7, 3)

oCell2.value=Numstring

' Друкуємо шапку на третьому аркуші

For j=0 to 3

oCell1=oSheet1.getCellByPosition(j, 1)

oCell3=oSheet3.getCellByPosition(j, Numstring)

volS=oCell1.string

oCell3.string=volS

next j

```

oCell3=oSheet3.getCellByPosition(4, Numstring)
oCell3.string="Количество"
oCell3=oSheet3.getCellByPosition(5, Numstring)
oCell3.string="Сумма (Грн.)"
oCell2=oSheet2.getCellByPosition(7, 3)
oCell2=oSheet2.getCellByPosition(7, 2)
oCell2.Value=0
oCell2=oSheet2.getCellByPosition(4, 4)
oCell2.Value=0
End sub

```

Призначте цей макрос кнопці На рахунок. Для цього у режимі конструювання проекту розкрийте вікно подій даної кнопки і, потім розкривши список макросів кнопкою з кількома крапками справа від події Призначити дію, виберіть: Order. Упевніться, що макрос працює вірно. Якщо це так, то на другому аркуші буде підведено підсумки і скориговані метадані як слід, а на третьому аркуші відбудеться підведення підсумків і надруковано шапку таблиці нового рахунку. Якщо не так, виконайте необхідні налагодження.

Випробуйте роботу проекту в цілому. Для цього збережіть увесь документ, закрийте його та знову відкрийте. Почніть з ініціалізації, потім випробуйте роботу списку, кнопок У корзину та На рахунок. Якщо робота проекту відбувається вірно, пред'явіть його викладачеві. Якщо не так, зробіть його повного налагодження.

Створіть звіт з результатів виконання практичного заняття, де дайте відповіді на наступні питання щодо UNO-технології:

1. Як адресуватися до одного з аркушів електронної таблиці?
2. Як адресуватися до комірки електронної таблиці? Як ввести у комірку число, текст, формулу?
3. Як адресуватися до діапазону комірок? Як призначити макрос кнопці?

Самостійні роботи

Самостійна робота № 1.

Тема: Суть засобів програмування офісних додатків операційних систем

Мета: з'ясувати суть програмування додатків операційних систем, розглянути їх еволюцію та отримати уявлення про основні складові цих засобів.

Завдання:

Створити звіт з самостійної роботи, де відобразити відповіді на питання:

1. У чому полягає суть програмування додатків операційних систем? Які основні етапи розвитку засобів програмування електронних документів відбулись з появленням потужних операційних систем ? Що таке макрос?
2. Як виконується доступ до інтегрованого середовища проектів **OOO IDE**?
3. Які засоби управління застосовуються у **OOO IDE**?
4. Що таке навігація по **OOO IDE** та які вона має складові?
5. Які основні функції складових навігації по **OOO IDE** має користувач?

Самостійна робота № 2.

Тема: Основи програмування мовою OOO BASIC.

Мета: З'ясувати суть програмування додатків мовою **OOO BASIC**.

Завдання на відпрацювання конспекту лекцій і літератури:

Створити звіт з самостійної роботи, де відобразити конспективні відповіді на питання:

1. Що являють собою основні складові мови **OOO BASIC**: змінні, константи процедури та функції?
2. Як здійснюється оголошення змінних?
3. У чому полягає суть процесу «оголосити, ввести, обробити

та вивести» в умовах офісного програмування.

4. Як виконується управління процесом виконання програм? У чому полягає різниця між підпрограмами та функціями? Що являють собою методи, властивості і події об'єктів?
5. Як визначено синтаксис використання методів і властивостей об'єктів у об'єктно-орієнтованому програмуванні (ООП)?
6. Що таке події і у чому полягає сенс їх застосувань при програмуванні електронних документів?

Самостійна робота № 3.

Тема: Організаційна структура засобів програмування електронних документів.

Мета: З'ясувати структуру засобів програмування електронних документів.

Завдання на відпрацювання конспекту лекцій і літератури:

Створити звіт, де відобразити конспективні відповіді на наступні питання:

1. Яка ієрархія визначає бібліотеки, модулі і макроси у середовищі OpenOffice.org?
2. Яку структуру мають бібліотеки, що зазначені вище?
3. Як відображається структура модулів і макросів, що представлені у бібліотеках?
4. У чому полягає методика користування бібліотеками, модулями і макросами?
5. За якою методикою виконується створення панельних і клавіатурних макросів?
6. Що являють собою області видимості для макросів?
7. Поняття моделі даних документу та двох концептуальних методів управління цією моделлю.
8. Поняття диспетчера і засобів користування ним.
9. Поняття глобальної команди та списку глобальних команд.

Самостійна робота № 4.**Тема: Основні поняття UNO- технології.**

Мета: поглибити поняття програмного інтерфейсу і сервісу та визначення їх елементів із довідкової системи.

Завдання на відпрацювання конспекту лекцій і літератури:

Створити звіт з самостійної роботи, де відобразити конспективні відповіді на наступні питання:

1. Як пов'язані між собою головні поняття технології UNO: інтерфейси і сервіси?
2. Якими функціями наділено сервіс DeskTop та його сервісні підпорядкування: поточних компонентів, поточних фреймів та завантаження документів.
3. Який вид має синтаксис UNO згідно маршруту до файлу?
4. Які загальні сервісні засоби операцій з файлами вживаються в умовах програмування електронних документів?
5. Яку структуру являє собою довідкова система он-лайн технології?
6. UNO та які її вона має основні функції і правила користування?

Самостійна робота № 5.**Тема: Робота з базами даних із середовища документів користувача. Засоби програмування взаємодій користувача з базами даних.**

Мета: засвоїти поняття щодо програмних засобів підключення до бази даних, доступу до даних та їх вилучення.

Завдання на відпрацювання конспекту лекцій і літератури:

Створити звіт з самостійної роботи, де відобразити конспективні відповіді на наступні питання:

1. Які засоби обміну даними на рівні бази даних — документ

- користувача підтримуються UNO-технологією?
2. Які програмні засоби реєстрації і підключення застосовуються до бази даних?
 3. Які програмні засоби підтримують доступ до таблиць бази даних. Як виконується програмування запитів із застосуванням елементів мови SQL?

Самостійна робота № 6.

Тема: Сервісні засоби UNO-технології для роботи з текстами.

Мета: поглибити знання відносно інтерфейсів, які забезпечують засоби текстових пересувань та обміну даними з програмою користувача.

Завдання на відпрацювання конспекту лекцій і літератури:

Створити звіт з самостійної роботи, де відобразити конспективні відповіді на наступні питання щодо он-лайн довідок інтерфейсів XTextCursor та XWordCursor:

www.openoffice.org/api/docs/common/ref/com/sun/star/text/XTextCursor.html,

www.openoffice.org/api/docs/common/ref/com/sun/star/text/XWordCursor.html:

1. Які функції виконують методи XTextCursor, що мають загальне визначення Collapse?
2. Які функції виконує кожний з методів цих інтерфейсів, котрий має загальне призначення пересувань курсору?
3. Які з методів XTextCursor використовуються як функції, а які виконують роль процедур?
4. Як дізнатися специфікацій виклику кожного з методів визначених вище інтерфейсів?
5. Що визначає специфікація bExpand у викликах методів інтерфейсів і яку роль вона відіграє у пересуваннях курсору

- та вставки тексту?
- б. Якими символами визначаються вхідні параметри методів?

Самостійна робота № 7.

Тема: Знайомство з елементами UNO-технології щодо програмування електронних таблиць

Мета: З'ясувати сутність програмування електронних таблиць, розглянути відповідні засоби та отримати уявлення про основні складові цих засобів.

Завдання на відпрацювання конспекту лекцій і літератури:

Створити звіт з самостійної роботи, де відобразити відповіді на питання:

1. У чому полягає суть програмування електронних таблиць?
2. Яку роль відіграють у процесі програмування електронних таблиць елементи управління?
3. Які програмні засоби застосовуються для відкриття та закриття електронних таблиць?
4. Як виконується програмний доступ до аркушів та комірок електронних таблиць?
5. Як виконується програмний доступ до діапазонів комірок електронних таблиць?
6. Які засоби застосовуються для програмної підтримки кількох таблиць?

Методи навчання

Пояснювально-ілюстративний, частинно-пошуковий (евристичний), репродуктивний, набуття знань, практичний, словесний, наочний, формування умінь і навичок, використання знань творча діяльність, закріплення, перевірка знань, умінь і навичок.

Форми контролю

№ з/п	Вид контролю	Методи контролю
1.	Поточний	усне опитування, перевірка якості виконання практичних завдань
2.	Періодичний	контрольні тестові роботи після завершення розділів,
3.	Самостійна робота	перевірка якості виконання завдань самостійної роботи
4.	Підсумковий контроль	іспит

Поточний контроль здійснюється у формі усного опитування, контрольних тестових робіт після завершення розділів, а також якості виконання завдань самостійної роботи та практичних завдань, тестування.

Підсумковий контроль здійснюється у формі іспиту

Відпрацювання пропущених занять здійснюється за узгодженням між студентом та викладачем.

Перезарахування результатів вивчення навчальної дисципліни відбувається у присутності комісії за умови доведення студентом/студенткою того рівня оцінки, на якій він/вона претендує.

Поточне тестування та самостійна робота							Залік	Сума
Розділ 1				Розділ 2				
T1	T2	T3	T4	T5	T6	T7	20	100
10	10	10	20	10	10	10		

Розподіл балів, які отримують здобувачі освіти

T1, T2, ..., T7 — теми розділів.

Розподіл балів за формами контролю.

- відвідування лекцій (пара) — 2; ...6 пар — до 12 балів
- виконання практичного завдання – 1-6; 11 практичних – 55 бал
- усне опитування — 1; ... 3 усних опитування — до 3 балів
- письмове опитування — до 10 балів
- виконання самостійного завдання — 1-2; 13 самостійних завдань — до 26 балів

Можлива кількість балів, набраних до підсумкової форми контролю:
80 балів

залік — 20 балів

Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	Для заліку
90-100	A	відмінно	зараховано
82-89	B	добре	
74-81	C		
64-73	D	задовільно	
60-63	E		
35-59	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0-34	F	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

Рекомендована література

Обов'язкова

1. OpenOffice.org 3.2 BASIC Guide. URL: https://wiki.openoffice.org/w/images/c/c1/BasicGuide_OOo3.2.0.pdf (дата звернення: 25.01.2024).
2. Andrew D. Pitonyak OpenOffice.org Macros Explained OOME Third Edition. URL: https://www.pitonyak.org/OOME_3_0.pdf (дата звернення: 25.01.2024).
3. OpenOffice.org 3.1 Developer's Guide. URL: https://wiki.openoffice.org/w/images/d/d9/DevelopersGuide_OOo3.1.0.pdf (дата звернення: 25.01.2024).
4. Learn OpenOffice.org Spreadsheet Macro Programming OOoBasic and Calc Automation Guide. URL: <https://www.pdfdrive.com/learn-openofficeorg-spreadsheet-macro-programming-ooobasic-and-calc-automation-a-fast-and-friendly-tutorial-to-writing-macros-and-spreadsheet-applications-e184846880.html> (дата звернення: 25.01.2024).

Допоміжна

5. The Apache OpenOffice API Project / Керівництво розробника: сайт. URL: <http://www.openoffice.org/api/> (дата звернення: 25.01.2024).
6. The OpenOffice.org Documentation Project: сайт. URL: <http://www.openoffice.org/api/>. (дата звернення: 25.01.2024).
7. Index of /api/docs/common/ref/com/sun/star URL: <http://www.openoffice.org/api/docs/common/ref/com/sun/star/> (дата звернення: 25.01.2024).

Інформаційні ресурси

1. Бібліотека Харківської державної академії культури.
2. Бібліотека Харківського національного університету В. Н. Каразіна.

3. Харківська державна наукова бібліотека імені В. Г. Короленка.

4. Українська бібліотечна енциклопедія URL:
<http://ube.nplu.org>

КОНТРОЛЬНІ ПИТАННЯ З ДИСЦИПЛІНИ

1. Поняття макросу. Пояснити призначення усіх команд у вікні діалогу **Макроси OpenOffice.org Writer**.
2. Пояснити методику протокольного створення макросів.
3. Пояснити призначення усіх команд меню **Файл** у середовищі **OOO BASIC**
4. Пояснити, методику контролю локальних змінних при налагодженні програм у середовищі **OOO BASIC**.
5. Типи даних **OOO BASIC**.
6. Дати перелік арифметичних та логічних операторів **OOO BASIC** та указати їх пріоритети.
7. Пояснити засоби організації циклів **Loop** у **OOO BASIC**
8. Пояснити засоби організації циклів **For** у **OOO BASIC**
9. Пояснити засоби організації циклів **While** у **OOO BASIC**
10. Пояснити засоби організації розвилок **If ... Then... Else** у **OOO BASIC**
11. Пояснити засоби організації розвилки **Select Case** у **OOO BASIC**
12. Пояснити технологію створення та застосування підпрограми **Function**.
13. Пояснити технологію створення та застосування підпрограми **Sub**.
14. Поняття об'єкту, властивості, методу, події.
15. Пояснити інструкції оголошення об'єктної змінної та присвоєння об'єкту об'єктної змінної.
16. Поняття **UNO**-технології: дати загальну характеристику інтерфейсів і сервісів.
17. Онлайн документація **UNO**-технології та її структура.

18. Загальний функціонал сервісу **Desktop** і глобальних змінних
19. **StarDesktop** та **ThisComponent**.
20. Засоби технології програмування баз даних: підключення до бази, перегляд таблиць та вилучення даних.
21. Поняття невидимого та видимого курсорів та їх основні функції з управління пересуванням по тексту.
22. Програмні засоби вставки та вилучення текстових фрагментів.
23. Програмні засоби пошуку елементів тексту.
24. Програмні засоби адресації до аркушів електронної таблиці
25. Програмні засоби адресації до комірок електронної таблиці.
26. Програмні засоби адресації до діапазонів комірок електронної таблиці. **Завдання**
27. Створити протокольний макрос, який при натисканні на кнопку виконує перевірку орфографії на українській мові.
28. Створити протокольний макрос, який при натисканні на кнопку виконує вставку у текст таблиці що має три стовпчика та десять рядків.
29. Створити протокольний макрос, що вставляє у текстовий документ графічний файл.
30. Створити шаблон поздоровлення зо святом, для вставки його у документ кнопкою панелі інструментів.
31. Створити протокольний макрос, що визначає особисті стилі форматування документа.
32. Створити фрагмент програми, що друкує на екрані монітора заголовки усіх фреймів **OpenOffice.org**.
33. Створити макрос, який виконує перегляд усіх властивостей та методів поточного документа.
34. Створити засіб перегляду у іншому документі усіх рядків таблиці бази даних, яка існує, у іншому документі за використанням елемента управління **Список**.
35. Створити засіб перегляду у текстовому документі усіх рядків з таблиці бази даних, яка існує, за використанням команди **Вид/Джерела даних**.

36. Створити засіб перегляду у середовищі електронної таблиці усіх рядків з таблиці бази даних, яка існує, за використанням команди **Вид/Джерела даних**.
37. Створити процедуру підключення до бази даних, яка існує.
38. Створити процедуру для перегляду усіх таблиць бази даних, яка існує.
39. Створити фрагмент програми для вилучення інформації з таблиці бази даних, яка існує, згідно з запитом **SQL**.
40. Створити макрос, що друкує з виключно з першого рядка текстового документу слова **Початок документу**.
41. Створити макрос, який друкує виключно з кінця останнього рядка текстового документу слово **Кінець**.
42. Створити макрос, що друкує у діалоговому вікні текст того абзацу, що виділено видимим курсором.
43. Створити макрос, який змінює текст абзацу з видимим курсором на інший текст, що збережено у строковій змінній.
44. Створити макрос, що надає можливість почергово переглянути всі абзаци тексту.
45. Створити макрос, який підраховує скільки абзаців мається у тексті від видимого курсору..
46. Створити макрос який видаляє абзац з видимим курсором.
47. Створити макрос, що пише рядок символів **Таблиця продаж** у комірку **A10** електронної таблиці.
48. Створити макрос, що пише нулі у діапазон **A1:A10** електронної таблиці.
49. Створити макрос, що пише у комірку **D5** формулу $=VLOOKUP(H3;Лист1.A3:D8;4)$.
50. Створити макрос, який очищає аркуш електронної таблиці.
51. Створити макрос, який діапазон **A1:H8** заповнює символами **0** і **1** у шаховому порядку.

Навчальне видання

ПРОГРАМУВАННЯ ОФІСНИХ ДОДАТКІВ

Програма та навчально-методичні матеріали до курсу
для здобувачів першого (бакалаврського) рівня вищої освіти зі
спеціальності 029 «Інформаційна, бібліотечна та архівна справа»
ОП «Інформаційна та документаційна діяльність»

Укладач:

Побіженко Ірина Олександрівна,
кандидат технічних наук, доцент, доцент кафедри цифрових
комунікацій та інформаційних технологій ХДАК

Друкується в авторській редакції

План 2024

Підписано до друку 24.04.2024 р. Формат 60x84/16
Гарнітура «Times». Папір для мн. ап. Друк ризограф.
Ум. Друк арк. 1,99. Обл.-вид. арк. 2,08. Тираж 100. Зам . №
ХДАК, 61057, м. Харків, Бурсацький узвіз, 4
Надруковано в лаб. множ. техніки ХДАК